

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of
the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- BLURRY OR ILLEGIBLE TEXT
- SKEWED/SLATED IMAGES
- COLORED PHOTOS
- BLACK OR VERY DARK BLACK AND WHITE PHOTOS
- UNDECIPHERABLE GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.



①9 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENTAMT

⑫ Off nlegungsschrift
⑩ DE 44 46 988 A 1

⑤1 Int. Cl.⁶:
G 01 R 31/3183

②1 Aktenzeichen: P 44 46 988.8
②2 Anmeldetag: 28. 12. 94
④3 Offenlegungstag: 27. 7. 95

DE 44 46 988 A 1

③0 Unionspriorität: ③2 ③3 ③1
28.12.93 JP 5-350145

⑦1 Anmelder:
Advantest Corp., Tokio/Tokyo, JP

⑦4 Vertreter:
Tauchner, P., Dipl.-Chem. Dr.rer.nat.; Heunemann,
D., Dipl.-Phys. Dr.rer.nat.; Rauh, P., Dipl.-Chem.
Dr.rer.nat.; Hermann, G., Dipl.-Phys. Dr.rer.nat.;
Schmidt, J., Dipl.-Ing.; Jaenichen, H., Dipl.-Biol.
Dr.rer.nat.; von Uexküll-Güldenband-Menzel, A.,
Dr.phil. (Ph.D.); Weinberger, R., Dipl.-Chem.Univ.
Dr.rer.nat.; Bublak, W., Dipl.-Chem. Univ.,
Pat.-Anwälte; Tremmel, H., Rechtsanw., 81675
München

⑦2 Erfinder:
Takano, Kazuo, Saitama, JP

⑤4 Schneller Testmustergenerator

⑤7 Es wird ein Testmustergenerator, welcher ein Testmuster mit hoher Geschwindigkeit ohne komplizierte Programmierung erzeugen kann, offenbart. Der Testmustergenerator weist auf: eine Parallelisierungsschaltung zum Umwandeln eines Instruktionssignals in serieller Form in n Instruktionssignale paralleler Form, eine Instruktionsverarbeitungsschaltung mit n-1 Instruktionsprozessoren zum Durchführen von Arithmetikoperationen auf der Basis von zwei Instruktionsdatensignalen aus den in der parallelen Form vorgegebenen Instruktionssignalen, und mehrere Registersätze, wovon jeder Satz mit einem gemeinsamen Taktsignal niedrigerer Geschwindigkeit versorgt wird, eine Arithmetikschialtung, die parallele Datensignale aus der Instruktionsverarbeitungsschaltung empfängt und Arithmetikoperationen mittels n darin enthaltener Arithmetikeinheiten durchführt und ferner ein Register enthält, welches die Datensignale in dem vorherigen Zyklus an die anderen Eingänge aller Arithmetikeinheiten liefert, und eine Multiplexschaltung zum Umwandeln der parallelen Datensignale aus der Arithmetikschialtung in eine serielle Form mit einer n-fach höheren Datenrate als die parallelen Datensignale auf der Basis eines schnelleren Taktes mit einer n-fach höheren Geschwindigkeit als das Taktsignal mit niedrigerer Geschwindigkeit.

DE 44 46 988 A 1

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

BUNDESDRUCKEREI 05. 95 508 030/407

14/28

Diese Erfindung betrifft einen Testmuster-generator, welcher in einem Halbleitertestsystem enthalten ist, und insbesondere einen Testmuster-generator, welcher einen sehr schnellen Betrieb zum Erzeugen eines Testmusters ohne das Erfordernis einer komplizierten Programmierung realisiert.

Im allgemeinen ist ein in einem Halbleitertestsystem verwendetes Testmuster zum Testen eines Halbleiterbausteins sehr lang und umfangreich. Ferner zeigt beispielsweise in einem Halbleitertestsystem ein Testmuster die Tendenz, denselben Abschnitt des Testmusters viele Male zu wiederholen. Des weiteren ist wegen des Anstiegs der Geschwindigkeit in den zu testenden neuesten Halbleiterbausteinen die Erzeugung solcher Testmuster mit hoher Geschwindigkeit erforderlich.

Ein zur Zeit verwendeter Testmuster-generator besteht im wesentlichen aus einer Akkumulatorarithmetikeinheit, welche ein langes und umfangreiches Testmuster durch Programmierung arithmetischer Instruktionen und einer Anzahl von Wiederholungen in einem Programmspeicher des Testmuster-generators erzeugt.

Fig. 10 stellt ein Beispiel einer konventionellen Arithmetikschialtung für den Testmuster-generator dar. Die Arithmetikschialtung umfaßt eine Steuerschialtung 3, eine Arithmetikeinheit 1 und ein Register 2. Die Steuerschialtung 3 besitzt einen Programmspeicher 31 und einen Taktgenerator 32. Ein aus dem Programmspeicher ausgelesener Befehl wird an einen Eingangsanschluß A der Arithmetikeinheit 1 geliefert. Der Ausgang der Arithmetikeinheit 1 ist mit dem Register 2 verbunden, dessen Ausgang mit einem Eingang B der Arithmetikeinheit 1 verbunden ist. Es wird auch ein Arbeitstakt aus dem Taktgenerator 32 an einen Takteingangsanschluß des Registers 2 angelegt. Das sich ergebende Ausgangssignal X wird aus dem Ausgang der Arithmetikeinheit 1 erhalten.

Fig. 11 stellt ein Betriebsbeispiel dar, welches ein Programmbeispiel 1 unter Verwendung der Arithmetikschialtung von Fig. 10 implementiert.

Zuerst ist ein Anfangswert "X=0" in der ersten Zeile des Programmspeichers 31 gespeichert. In der nächsten Zeile ist die Instruktion "Wiederhole 11 $X=X+1$ " gespeichert, welche bedeutet, eine Addition von Eins (1) elfmal zu wiederholen. In der dritten Zeile des Programmspeichers 31 ist " $X=X$ " vorgeschrieben, damit der Ausgangswert X unverändert bleibt (als X beibehalten wird). Somit bestehen, wie vorstehend beschrieben, die Instruktionsdatensignale für die Arithmetikeinheit aus einem Satz eines numerischen Wertes, eines Vorzeichens und eines Befehls.

Daher wird in einem ersten Betriebszyklus der Befehl $X=0$ aus dem Programmspeicher 31 an den Anschluß A der Arithmetikeinheit 1 ausgegeben. Die Arithmetikeinheit 1 führt eine Arithmetikoperation aus. Da der Befehl aus dem Speicher 31 gleich $X=0$ lautet, empfängt die Arithmetikeinheit 1 "0" als ihren Eingangswert, ohne den Eingangswert am anderen Eingangsanschluß B zu berücksichtigen. Somit wird der Wert "0" als ein Ausgangssignal X der Arithmetikeinheit 1 erhalten.

Als nächstes wird der Arbeitstakt an das Register 2 angelegt. Als Antwort auf den Takt speichert das Register 2 den Wert 0 und sein Ausgangssignalwert wird zu 0. Dieser numerische Wert wird an den anderen Eingangsanschluß B der Arithmetikeinheit 1 angelegt. Da an den Eingangsanschluß A in diesem Zyklus $X=+1$ eingegeben wird und in den Eingangsanschluß B der

Wert 0 eingegeben wird, wird die arithmetische Operation " $0+1$ " von der Arithmetikeinheit 1 ausgeführt. Die Arithmetikeinheit 1 erzielt ein Ausgangssignal $X=1$ als Ergebnis. In gleicher Weise werden nur ein numerischer Wert und ein Vorzeichen an den anderen Eingangsanschluß B der Arithmetikeinheit 1 angelegt.

Im dritten Betriebszyklus wird ein Befehl " $+1$ " dauernd an den Anschluß A ausgegeben, und die Arithmetikoperation von plus Eins für den vorherigen Ausgangssignalwert $X=1$ (Anschluß B) ausgeführt. Daher wird ein Ausgangssignalwert $X=2$ als Ergebnis am Ausgang der Arithmetikeinheit 1 erhalten. Auf diese Weise wird die Arithmetikoperation der " $+1$ "-Addition 11mal (elfmal) ausgeführt.

In der vorstehenden Betriebsprozedur in dem konventionellen Testmuster-generator ändert der Ausgang des Registers 2 seinen Zustand mit dem Empfang des Arbeitstaktes und die Arithmetikeinheit 1 führt eine Arithmetikoperation auf der Basis des Ausgangssignalswertes des Registers 2 aus. Nach der für diese Operation benötigten Zeit wird die Lieferung des Arbeitstaktes für den nächsten Taktzyklus möglich. Dieser Arbeitstaktzyklus ergibt den kürzesten Ablauf der Mustererzeugungsprozedur. Demzufolge entspricht das Zeitintervall des Taktes der höchsten Berechnungsgeschwindigkeit in diesem konventionellen Testmuster-generator.

Fig. 12 stellt ein weiteres Beispiel einer Arithmetikschialtung in dem konventionellen Testmuster-generator dar. In diesem Beispiel besteht das Verfahren zur Erhöhung der Testmustererzeugungsgeschwindigkeit einfach darin, mehrere Arithmetikeinheiten in paralleler Form zu verwenden. Es müssen jedoch immer noch Instruktionen im voraus für jede Arithmetikeinheit vorbereitet werden, um diesen Parallelbetrieb zu ermöglichen.

In dem Beispiel von Fig. 12 sind vier Arithmetikeinheiten 11 bis 14, vier Register 21 bis 24 und vier Programmspeicher 41 bis 44 in paralleler Form bereitgestellt. Jeder aus der Arithmetikeinheit, dem Register und dem Programmspeicher bestehende Satz ist identisch zur Struktur von Fig. 10. Die Register 21 bis 24 empfangen ein gemeinsames Taktsignal von einem Taktgenerator 45. Die Ausgangssignale X_0, X_1, X_2 und X_3 der Arithmetikeinheiten 11 bis 14 werden an eine Multiplexschaltung 5 geliefert, welche ebenfalls mit einem Taktsignal von dem Taktgenerator 45 versorgt wird. Die Taktrate für die Multiplexschaltung in diesem Beispiel sollte viermal schneller als die des Beispiels von Fig. 10 sein, da die vier Arithmetikeinheiten 11 bis 14 parallel angeordnet sind.

Ein Konfigurationsbeispiel der Multiplexschaltung 5 ist in Fig. 6 dargestellt. Gemäß Darstellung in Fig. 6 umfaßt die Multiplexschaltung einen Multiplexer 502 und einen Zähler 501. Der Zähler 501 empfängt den Takt aus dem Taktgenerator 45 und gibt ein 2-Bit-Signal für den Multiplexer 502 aus. Das 2-Bit-Signal arbeitet als Selektionssignal für den Multiplexer 502. Die Ausgangssignale X_0, X_1, X_2 und X_3 der Arithmetikeinheiten 11 bis 14 werden von dem Multiplexer 502 mit dem Zeittakt des Selektionssignals aus dem Zähler 501 multiplexiert, und das selektierte Ausgangssignal wird an dem Ausgangsanschluß der Multiplexschaltung 5 als ein sich ergebendes Ausgangssignal X mit einer vierfach schnelleren Geschwindigkeit als die Datensignale aus den Arithmetikeinheiten 11 bis 14 bereitgestellt.

Fig. 7 ist ein Zeitdiagramm, welches eine grundlegende Betriebsweise der Multiplexschaltung von Fig. 6 darstellt. Nach allgemeiner Kenntnis im Fachgebiet können

n parallele Datensignale, die an einen Multiplexer angelegt sind, in ein seriellles Datensignal mit einer n-fach schnelleren Geschwindigkeit als jedes der parallelen Datensignale umgewandelt werden. In Fig. 7 werden parallel an den Multiplexer 502 angelegte Eingangssignale X_0 , X_1 , X_2 und X_3 mit Daten a, b, c und d in ein Ausgangssignal X mit den Daten a, b, c und d in Serie umgewandelt, welches viermal schneller als jedes der Eingangssignale ist.

Fig. 13 ist eine schematische Ansicht, welche ein Beispiel der Betriebsweise der Arithmetikeinheit von Fig. 12 zur Implementierung einer Testmustererzeugung auf der Basis eines Programmbeispiels 1 darstellt. Gemäß Darstellung in einer Tabelle von Fig. 13A enthält das Programmbeispiel 1 einen in dem Programmspeicher 41 gespeicherten Anfangswert, welcher an die Schalteinrichtung 11 angelegt wird. Der Anfangswert für die Arithmetikeinheit 11 ist:

$$X_0 = 0.$$

Der in dem Programmspeicher 42 gespeicherte Anfangswert für die Arithmetikeinheit 12 ist:

$$X_1 = 1.$$

Der in dem Programmspeicher 43 gespeicherte Anfangswert für die Arithmetikeinheit 13 ist:

$$X_2 = 2.$$

Der in dem Programmspeicher 44 gespeicherte Anfangswert für die Arithmetikeinheit 14 ist:

$$X_3 = 3.$$

Diese Anfangswerte werden gemäß Darstellung in Fig. 13B an die entsprechenden Eingangsanschlüsse der Arithmetikeinheiten 11 bis 14 im ersten Zyklus angelegt. Demzufolge repräsentieren gemäß Darstellung in Fig. 13B die Ausgangsdatsignale der Arithmetikeinheiten 11 bis 14 im ersten Zyklus denselben Wert 0, 1, 2 und 3.

Der nächste auch im Programmspeicher 41 gespeicherte Wert für die Arithmetikeinheit 11 ist:

$$\text{Wiederhole } 2 X_0 = X_0 + 4.$$

Der nächste auch im Programmspeicher 42 gespeicherte Wert für die Arithmetikeinheit 12 ist:

$$\text{Wiederhole } 2 X_1 = X_1 + 4.$$

Der nächste auch im Programmspeicher 43 gespeicherte Wert für die Arithmetikeinheit 13 ist:

$$\text{Wiederhole } 2 X_2 = X_2 + 4.$$

Der nächste auch im Programmspeicher 44 gespeicherte Wert für die Arithmetikeinheit 14 ist:

$$\text{Wiederhole } X_3 = X_3 + 4.$$

Demzufolge addiert im zweiten Zyklus jede Arithmetikeinheit 11 bis 14 vier (4) zu dem Datensignal des ersten Zyklus hinzu, da jedes Register 21 bis 24 gemäß Darstellung in Fig. 13C die vorherigen Datensignale an andere Eingangsanschlüsse der Arithmetikeinheiten zurückgibt. Folglich repräsentieren im zweiten Zyklus die

Ausgangsdatsignale der Arithmetikeinheiten 11 bis 13 gemäß Darstellung in Fig. 13C die Werte 4, 5, 6 und 7.

Im dritten Zyklus wird, da die Instruktion für die Arithmetikeinheit 11 dieselbe wie im vorherigen Zyklus ist, d. h., "Wiederhole $2 X_0 = X_0 + 4$ ", was die Wiederholung der Addition von vier (4) erfordert, der Ausgangsdatsignalwert der Arithmetikeinheit 11 gemäß Darstellung in Fig. 13D zu 8. In ähnlicher Weise wird im dritten Zyklus, da die Instruktionen für die Arithmetikeinheiten 12 und 13 ebenfalls dieselben wie die im vorherigen Zyklus sind, d. h., "Wiederhole $2 X_1 = X_1 + 4$ " bzw. "Wiederhole $2 X_2 = X_2 + 4$ ", der Ausgabedatsignalwert der Arithmetikeinheiten 12 und 13 gemäß Darstellung in Fig. 13D zu 9 bzw. 10.

Der Ausgabedatsignalwert der Arithmetikeinheit 14 repräsentiert im dritten Zyklus 10, da die Instruktion im Programmspeicher 44 angibt:

$$X_3 = X_3 + 3.$$

Gemäß vorstehender Beschreibung erhöht sich im Vergleich zu dem herkömmlichen Beispiel in Fig. 10 und 11 die Geschwindigkeit der Testmustererzeugung um die Anzahl der parallel bereitgestellten Arithmetikeinheiten (4fach im Beispiel von Fig. 12). Da jedoch das in den Programmspeichern 41 bis 44 zu speichernde Programm gemäß dem obenstehenden Beispiel komplizierter wird, ist somit eine Programmentwicklungsprozedur für dieses Beispiel zeitaufwendig und mühsam.

Es ist daher eine Aufgabe der vorliegenden Erfindung, eine Lösung für die vorstehenden, in den konventionellen Testmustergeräten gefundenen Probleme bereitzustellen. Diese Aufgabe wird durch die Erfindung gelöst. Die vorliegende Erfindung stellt einen Testmustererzeuger bereit, welcher ein schnelles Testmuster ohne das Erfordernis eines komplizierten Programmierprozesses erzeugen kann.

Ein Testmustererzeuger umfaßt eine Parallelisierungsschaltung zum Umwandeln eines Instruktionssignals in eine parallele Form und eine Instruktionsverarbeitungsschaltung, welche n-1 Instruktionssprozessoren aufweist, die Arithmetikoperationen ausführen, wovon jede Operation auf zwei Instruktionssignalen aus n parallelen Instruktionssignalen basiert. Für jeden Zyklus des Arbeitstaktes sind n Register in der Instruktionsverarbeitungsschaltung zum Verschieben der Instruktionsdatensignale zu der nächsten Stufe bereitgestellt. Der Testmustererzeuger enthält ferner eine Arithmetiksenschaltung zum Aufnehmen der Instruktionsdatensignale aus der Instruktionsverarbeitungsschaltung und zum Ausführen von Akkumulatorfunktionen durch n Arithmetikeinheiten. Die Arithmetiksenschaltung enthält ein Register, welches in dem vorherigen Zyklus aufgetretene Datensignale gemeinsam an die Eingänge aller Arithmetikeinheiten liefert. Eine Multiplexerschaltung ist dafür bereitgestellt, die Eingangsdatsignale aus der Arithmetiksenschaltung in ein seriellles Datensignal mit n-mal schnellerer Geschwindigkeit als jedes der Eingangsdatsignale umzuwandeln.

Der vorliegenden Erfindung entsprechend kann die Arbeitsgeschwindigkeit des Instruktionssignals aus einem Programmspeicher n-fach höher als die Arbeitsgeschwindigkeit jedes Instruktionssignals in der Instruktionsverarbeitungsschaltung eingestellt werden. Auch die Geschwindigkeit der verarbeiteten Datensignale aus der Multiplexerschaltung kann gegenüber der Arbeitsgeschwindigkeit der Register und des Instruktionssprozessors in der Instruktionsverarbeitungsschaltung oder den

Arithmetikeinheiten in den Arithmetikschaltungen n-fach multipliziert werden. Daher sind bei der vorliegenden Erfindung n-fach schnellere arithmetische Operationen als die konventionellen möglich. Da ferner der herkömmliche Instruktionsschritt bei einem Programmierprozeß der Testmustererzeugung verwendet werden kann, erfordert die vorliegende Erfindung keinerlei komplizierte Programmierprozeduren.

Von den beigefügten Zeichnungen stellen dar:

Fig. 1 ein Blockschaltbild, das eine Schaltungskonfiguration des Testmustergenerators der vorliegenden Erfindung zeigt;

Fig. 2 eine Struktur eines Beispiels erfindungsgemäßer Arithmetikschaltungen;

Fig. 3 arithmetische Funktionen einer Arithmetikeinheit der vorliegenden Erfindung und zugeordnete, an die Arithmetikeinheit angelegte Befehle;

Fig. 4 ein Beispiel einer in der Parallelisierschaltung von Fig. 1 enthaltenen Schaltungskonfiguration;

Fig. 5 ein Zeitdiagramm, welches eine Betriebsweise der Parallelisierschaltung von Fig. 4 zeigt;

Fig. 6 ein Beispiel einer Schaltungsstruktur für eine Multiplexerschaltung;

Fig. 7 ein Zeitdiagramm, welches eine Betriebsweise der Multiplexerschaltung von Fig. 6 zeigt;

Fig. 8 ein Zeitdiagramm, welches ein Beispiel einer erfindungsgemäßen Betriebsweise zeigt;

Fig. 9 eine graphische Ansicht, welche ein weiteres Beispiel einer erfindungsgemäßen Betriebsweise zeigt;

Fig. 10 ein Beispiel einer in dem konventionellen Testmustergenerator enthaltenen Arithmetikschaltung;

Fig. 11 eine graphische Ansicht, welche eine Betriebsweise der Arithmetikschaltung von Fig. 10 zeigt;

Fig. 12 ein weiteres Beispiel einer in dem konventionellen Testmustergenerator enthaltenen Arithmetikschaltung; und

Fig. 13 eine graphische Ansicht, welche eine Betriebsweise der Arithmetikschaltung von Fig. 12 zeigt.

Fig. 1 ist ein Blockschaltbild, das eine Ausführungsform des Testmustergenerators der vorliegenden Erfindung zeigt. In diesem Beispiel wird die Geschwindigkeit der Testmustererzeugung um das 4fache gegenüber dem konventionellen Testmustergenerator erhöht, da vier parallele Schaltungen eingesetzt werden. Gemäß Fig. 1 enthält der Testmustergenerator einen Parallelisierschaltung 6, eine Steuerschaltung 7, eine Instruktionsverarbeitungsschaltung 100, eine Arithmetikschaltung 200 und eine Multiplexerschaltung 5.

Die Steuerschaltung 7 weist einen Programmspeicher 71, welcher eine Serie von Instruktionsdatensignalen speichert, und einen Taktgenerator 72 auf, welcher einen Arbeitstakt und einen $1/n$ -Arbeitstakt erzeugt. Der Arbeitstakt ist n-mal schneller als der $1/n$ -Arbeitstakt (im Beispiel von Fig. 1 gilt: $n=4$). Die Parallelisierschaltung 6 wandelt die Instruktionssignale ix aus dem Programmspeicher 71 in vier parallele Instruktionssignale i0, i1, i2 und i3 um und ordnet die Instruktionssignale vier Eingängen der Instruktionsverarbeitungsschaltung 100 zu.

Die Instruktionsverarbeitungsschaltung 100 enthält Instruktionsprozessoren 101, 102 und 103 und Register 111 bis 122. Die Register 111 bis 122 werden mit dem $1/4$ -Arbeitstakt aus dem Taktgenerator 72 versorgt. Gemäß Darstellung in Fig. 1 werden die an den Eingangsanschlüssen der Instruktionsverarbeitungsschaltung 100 empfangenen vier Instruktionssignale von den Instruktionsprozessoren und Registern in vier Signalpfaden verarbeitet und als Instruktionen i0c, i1c, i2c und i3c an

vier Ausgangsanschlüssen ausgegeben. Die Arithmetikschaltung 200 enthält vier Arithmetikeinheiten 201 bis 204, deren jeweiliger Eingangsanschluß die Instruktionen i0c, i1c, i2c und i3c aufnimmt, und ein Register 205, welches mit dem Ausgang der Arithmetikeinheit 204 verbunden ist. Jeder von den anderen Eingangsanschlüssen der Arithmetikeinheiten 201 bis 204 wird von einem gemeinsamen Ausgangssignal des Registers 205 versorgt.

Die Multiplexerschaltung 5 empfängt Instruktionssignale X_0 , X_1 , X_2 und X_3 aus der Arithmetikschaltung 200. Ein Beispiel der Schaltungskonfiguration und der Betriebsweise der Multiplexerschaltung 5 sind in den Fig. 6 und 7, wie vorstehend erläutert, dargestellt. Die Multiplexerschaltung 5 wandelt die parallelen Eingangssignale um und gibt das Instruktionssignal X in der seriellen Form an seinem Ausgang aus.

Fig. 4 zeigt ein Beispiel einer in dem Parallelisierschaltung 6 enthaltenen Schaltungskonfiguration. Der Hauptzweck der Parallelisierschaltung 6 besteht in der Umwandlung eines seriellen Signals in ein paralleles Signal. Es wird nämlich das Instruktionssignal ix aus dem Programmspeicher 71, welches ein serielles Signal ist, durch die Parallelisierschaltung 6 in die vier parallelen Befehlssignale i0, i1, i2 und i3 umgewandelt. Die Parallelisierschaltung 6 enthält einen Zähler 401, einen Dekoder 402, einen Satz Register 411 bis 414 und einen weiteren Satz Register 421 bis 424.

Der Zähler 401 wird mit dem von dem Taktgenerator 72 erzeugten Arbeitstakt versorgt. Das Ausgangssignal des Zählers 401, welches beispielsweise ein 2-Bit-Signal ist, wird an den Dekoder 402 geliefert. Der Dekoder 402 erzeugt 4 dekodierte Signale auf der Basis dieses 2-Bit-Signals, und die 4 dekodierten Signale werden an die entsprechenden Taktanschlüsse der Register 411, 412, 413 und 414 geliefert. Daher speichert jedes der Register 411 bis 414 das Instruktionssignal ix während der Zeit, welche 4-mal länger als die Periode des Arbeitstaktes ist. Jedes Instruktionssignal, welches in den entsprechenden Registern 411 bis 414 gespeichert ist, wird synchronisiert mit dem dekodierten Signal ($1/4$ -Arbeitstakt) zu den Registern 421, 422, 423 und 424 verschoben, und als die parallelen Instruktionssignale i0, i1, i2 und i3 ausgegeben.

Fig. 5 ist ein Zeitdiagramm, welches eine Betriebsweise des Parallelisierschaltung 6 darstellt. Das Instruktionssignal ix weist ein Datensignal (Daten) a, b, c und d in serieller Form auf, welches eine Signalarate gleich dem Arbeitstakt besitzt. Die Datensignale a, b, c und d werden in den entsprechenden Registern 411, 412, 413 und 414 getrennt und zeitlich im nächsten Zyklus des dekodierten Signal durch die Register 421, 422, 423 und 424 justiert. Daher werden die Datensignale a, b, c und d in dem Instruktionssignal ix zu den parallelen Instruktionssignalen i0, i1, i2 und i3, wovon jedes den entsprechenden Datensignalwert a, b, c und d aufweist. Die Signalarate der parallelen Instruktionssignale i0, i1, i2 und i3 ist viermal höher als die des Instruktionssignals ix.

Die Instruktionsverarbeitungsschaltung 100 verarbeitet die Instruktionssignale i0, i1, i2 und i3 aus der Parallelisierschaltung 6. Die Instruktionssignale i0 und i1 werden an den zwei Eingangsanschlüssen aufweisenden Instruktionsprozessor 101 geliefert. Der Instruktionsprozessor 102 empfängt das Instruktionssignal i2 über das Register 117, während der Instruktionsprozessor 103 das Instruktionssignal i3 über die Register 120 und 121 empfängt. Ein Beispiel der detaillierteren Struktur der in der vorliegenden Erfindung zu verwendenden Instruk-

tionsprozessoren 101 bis 103 ist in Fig. 2 dargestellt. Der Instruktionsprozessor 101 enthält Instruktionsdekoder 301 und 302, ein UND-Gatter 304, eine Arithmetikeinheit 303 und ein ODER-Gatter 305.

Das Instruktionssignal $i0$ wird sowohl an einen Eingangsanschluß des UND-Gatters 304 als auch an den Instruktionsdekoder 391 angelegt. Der Ausgang des UND-Gatters 304 ist mit einem Eingangsanschluß A der Arithmetikeinheit 303 verbunden. Das Instruktionssignal $i0$ enthält zusätzlich zu einem numerischen Wert ein Vorzeichen und auch einen Befehlssatz, welcher eine Substitution anzeigt. Der Instruktionsdekoder 301 detektiert den "Vorzeichen"-Befehl und den "Substitutions"-Befehl in dem Instruktionssignal $i0$. Das Vorzeichen (+/-), das von dem Instruktionsdekoder 301 detektiert wird, wird an die Arithmetikeinheit 303 weitergegeben. Ebenso wird der detektierte "Substitutions"-Befehl an das ODER-Gatter 305 weitergegeben.

Das Instruktionssignal $i1$ wird an den anderen Eingangsanschluß B der Arithmetikeinheit 303 angeschlossen. Der Instruktionsdekoder 302 detektiert ebenfalls den "Vorzeichen"-Befehl und den "Substitutions"-Befehl in dem Instruktionssignal $i1$. Der detektierte Vorzeichen (+/-)-Befehl wird an die Arithmetikeinheit 303 weitergegeben. Der detektierte "Substitutions"-Befehl wird an den anderen Eingangsanschluß des ODER-Gatters 305 weitergegeben. Des weiteren wird der detektierte "Substitutions"-Befehl in dem Instruktionssignal $i1$ invertiert und an das UND-Gatter 304 geliefert. Auf die Weise wird dann, wenn der "Substitutions"-Befehl in dem Instruktionssignal $i1$ enthalten ist, das Instruktionssignal $i0$ ignoriert. Die Arithmetikeinheit 303 arbeitet ausschließlich auf der Basis des Instruktionssignals $i1$ und gibt ein "Ergebnis $i1$ " aus, welches den numerischen Wert und das Vorzeichen einschließt. Das ODER-Gatter 305 gibt den im nächsten Schritt zu verwendenden "Substitutions"-Befehl aus.

Die arithmetischen Funktionen der Arithmetikeinheiten 101 bis 103 sind in Fig. 3 dargestellt. Die arithmetischen Befehle für die Arithmetikeinheit werden in Addition, Subtraktion, Substitution und keine Berechnung klassifiziert. Wenn die Instruktion $i0$ und die Instruktion $i1$ gegeben sind, wird das "Ergebnis $i1$ " der Arithmetikoperation gemäß Darstellung in der Tabelle von Fig. 3 erreicht.

Gemäß Darstellung in Fig. 1 ist der Ausgang der Arithmetikeinheit 101 mit dem Register 114 verbunden. Mit dem Takt des 1/4-Arbeitstaktes wird das Ausgangssignal des Registers 114 zu $i1a$. Gleichzeitig liefern die Register 111, 117 und 120 Instruktionsergebnisse $i0a$, $i2a$ und $i3$, welche den Instruktionssignalen $i0$, $i2$ und $i3$ entsprechen, an die nächste Stufe. Die Arithmetikeinheit 102 wird mit den Instruktionssignalen $i0a$ und $i2a$ versorgt und führt die Arithmetikoperation in derselben Weise wie in der Arithmetikeinheit 101, wie vorstehend erläutert, aus.

Das Ergebnis der Arithmetikoperation in der Arithmetikeinheit 102 wird mit dem 1/4-Arbeitstakt von dem Register 118 übernommen. Somit wird das Ausgangssignal des Registers 118 zu einem Instruktionssignal $i2b$. Gleichzeitig verschieben die Register 112, 115 und 121 die Instruktionen aus den Registern 111, 114 und 120, welche nun Instruktionssignale $i0b$, $i1b$ und $i3b$ gemäß Darstellung in Fig. 1 sind. In ähnlicher Weise führt die Arithmetikeinheit 103 auf der Basis der angelegten Instruktionssignale $i2b$ und $i3b$ die Arithmetikoperation aus, welche ein Instruktionssignal $i3c$ über das Register 122 ergibt. Die Register 113, 116 und 119 senden eben-

falls Instruktionssignale $i0c$, $i1c$ bzw. $i2c$ aus. Wie vorstehend beschrieben, enthält die Instruktionsverarbeitungsschaltung 100 der vorliegenden Erfindung drei ($n-1$) Arithmetikeinheiten 101, 102 und 103 für vier (n) Instruktionssignale.

Die Arithmetikschialtung 200 empfängt die Instruktionssignale $i0c$, $i1c$, $i2c$ und $i3c$ an entsprechenden Eingängen der Arithmetikeinheiten 201, 202, 203 und 204 gemäß Darstellung in Fig. 1. Das Register 205 liefert das Datensignal X_R aus der Arithmetikeinheit 204 im vorherigen Zyklus an die anderen Eingänge der Arithmetikeinheiten 201, 202, 203 und 204. Die Arithmetikschialtung 200 führt eine Arithmetikoperation unter dieser Konfiguration aus und erzeugt Ausgabedatensignale X_0 , X_1 , X_2 und X_3 , welche von der Multiplexschaltung 5 empfangen werden.

Die Multiplexschaltung 5 wandelt die parallelen Datensignale X_0 , X_1 , X_2 und X_3 aus der Arithmetikschialtung 200 in ein serielles Datensignal X um. Das Datensignal X weist eine viermal schnellere Geschwindigkeit auf als jedes der parallelen Datensignale X_0 , X_1 , X_2 und X_3 , da in der Multiplexschaltung 5 der Arbeitstakt verwendet wird. Gemäß vorstehender Beschreibung ist die Multiplexschaltung 5, wie in dem Beispiel von Fig. 6 dargestellt, konfiguriert. Fig. 7 ist ein Zeitdiagramm, das eine Betriebsweise der Multiplexschaltung 5 darstellt, so wie es auch vorstehend beschrieben wurde.

Fig. 8 stellt ein Zeitdiagramm dar, welches eine Gesamtbetriebsweise des Testmustererzeugers von Fig. 1 auf der Basis eines oben rechts in Fig. 8 dargestellten Programmbeispiels zeigt. In diesem Beispiel sind die ersten Werte der parallelen Instruktionssignale $i0$, $i1$, $i2$, $i3$ aus der Parallelisierschialtung 6 gleich 0, +1, +1, +1. Das Ergebnis der Arithmetikoperation ($0+1$) in der Arithmetikeinheit 101 ist "1", welches von dem Register 114 in dem zweiten Zyklus des 1/4 Arbeitstaktes gespeichert wird. Daher zeigen die Instruktionssignale $i0a$, $i1a$, $i2a$ und $i3a$ im zweiten Zyklus 0, 1, 1 und 1.

Da beide Eingangssignale für die Arithmetikeinheit 102 "1" anzeigen, zeigt deren Ausgang "2", was von dem Register 118 im dritten Zyklus gespeichert wird. Daher zeigen die Instruktionssignale $i0b$, $i1b$, $i2b$ und $i3b$ im zweiten Zyklus 0, 1, 2 und 1. Das Ergebnis der Arithmetikoperation in der Arithmetikeinheit 103 ist "3", da die Eingangssignale dazu "2" und "1" sind, was von dem Register 122 im vierten Zyklus gespeichert wird. Daher zeigen die Instruktionssignale $i0c$, $i1c$, $i2c$ und $i3c$ im vierten Zyklus 0, 1, 2 und 3.

Die Instruktionssignale $i0c$, $i1c$, $i2c$ und $i3c$ werden von der Arithmetikschialtung 200 aufgenommen, in welcher das Register 205 das Datensignal X_R in dem Instruktionssignal $i3c$ im vorherigen Zyklus gemeinsam an die Arithmetikeinheiten 201 bis 204 für die Akkumulationsoperation in den Arithmetikeinheiten 201 bis 204 liefert. Daher werden beispielsweise im dritten Zyklus von Fig. 8E, da der Wert "3" in dem zweiten Zyklus von dem Register 205 gespeichert und an die Arithmetikeinheiten 201 bis 204 geliefert wird, die Ausgabedatensignale X_0 , X_1 , X_2 und X_3 zu 4, 5, 6 und 7.

Dieses kommt daher, da die anderen Eingänge der Arithmetikeinheiten 201 bis 204 mit den Instruktionssignalen $i0c$, $i1c$, $i2c$ und $i3c$, die 1, 2, 3 und 4 anzeigen, versorgt werden. Die parallelen Ausgabedatensignale X_0 , X_1 , X_2 und X_3 werden in ein serielles Datensignal X umgewandelt, welches durch die Multiplexschaltung 5 eine Wiederholungsrate viermal schneller als die von jedem parallelen Datensignal aufweist. Demzufolge enthält das serielle Datensignal X in dem obigen Beispiel

ein Datensignalfolge von "1, 2, 3, 4, 5, 6, 7, 8...".

Fig. 9 stellt ein weiteres Betriebsbeispiel der erfindungsgemäßen Testmustererzeugung dar. In Fig. 9 beginnt die Operation mit einem Anfangswert "0". Nach der Implementation eines viermaligen Additionsbefehls wird der nächste Anfangswert auf "8" gesetzt. Dann wird ein sechsmaliger Subtraktionsbefehl gesetzt. Auch in dieser Situation, bei der ein Substitutionsbefehl während der Operation vorgesehen ist, wird die Testmustererzeugung ohne jede Hindernisse ausgeführt, d. h., ohne daß irgendwelche nachteiligen Effekte von Arithmetikoperationen in den vorherigen Zyklen empfangen werden.

Gemäß vorstehender Beschreibung kann bei der vorliegenden Erfindung die Arbeitsgeschwindigkeit des Instruktionssignals ix viermal (n-mal) höher als die Arbeitsgeschwindigkeit jedes Instruktionssignals i0, i1, i2 und i3 in der Instruktionsverarbeitungsschaltung 100 gesetzt werden. Auch die Geschwindigkeit der von der Multiplexschaltung 5 verarbeiteten Datensignale kann mit dem vierfachen (n-fachen) der Arbeitsgeschwindigkeit der Register und Arithmetikeinheiten in den vorherigen Schaltungen multipliziert werden. Mit anderen Worten, im Vergleich zur konventionellen Arbeitsgeschwindigkeit ist bei der vorliegenden Erfindung eine vierfach (n-fach) schnellere Arithmetikoperation verfügbar. Da desweiteren die konventionellen Instruktionsschritte bei der Programmierung der Testmustererzeugung angewendet werden können, erfordert die vorliegende Erfindung keinerlei komplizierte Programmierprozeduren.

Patentansprüche

1. Testmusterergenerator mit:
 - einer Parallelisierungsschaltung zum Umwandeln eines Instruktionssignals serieller Form in n Instruktionssignale paralleler Form;
 - einer Instruktionsverarbeitungsschaltung, welche n-1 Instruktionsprozessoren zum Ausführen von Arithmetikoperationen auf der Basis von zwei Instruktionsdatensignalen aus den in paralleler Form vorgegebenen Instruktionssignalen und mehrere Registersätze aufweist, wovon jeder Satz von einem gemeinsamen Taktsignal mit niedrigerer Geschwindigkeit versorgt wird;
 - einer Arithmetikschialtung, welche die parallelen Datensignale aus der Instruktionsverarbeitungsschaltung aufnimmt und Arithmetikoperationen mittels n darin befindlicher Arithmetikeinheiten ausführt, wobei die Arithmetikschialtung ein Register enthält, welches die Datensignale in dem vorherigen Zyklus an die anderen Eingänge aller Arithmetikeinheiten liefert; und
 - einer Multiplexschaltung zum Umwandeln der parallelen Datensignale aus der Arithmetikschialtung in eine serielle Form mit n-mal schnellerer Datenrate als die parallelen Datensignale auf der Basis eines schnelleren Taktsignals mit einer n-mal schnelleren Geschwindigkeit als der Takt mit der niedrigeren Geschwindigkeit.
2. Testmusterergenerator nach Anspruch 1, wobei der Instruktionsprozessor aufweist:
 - einen ersten Eingangsanschluß und einen zweiten Eingangsanschluß;
 - eine Arithmetikeinheit zum Ausführen einer Arithmetikoperation mit an den Eingangsanschlüsse angelegten Instruktionsdatensignalen;

ein mit dem ersten Eingangsanschluß verbundenes UND-Gatter;

einen ersten Instruktionsdekoder zum Detektieren eines Vorzeichens und eines Substitutionsbefehls aus den an den ersten Eingangsanschluß angelegten Instruktionsdatensignalen;

ein mit dem detektierten Substitutionsbefehlsignal des ersten Instruktionsdekoders verbundenes ODER-Gatter;

einen zweiten Instruktionsdekoder zum Detektieren eines Vorzeichens und eines Substitutionsbefehls aus den an den zweiten Eingangsanschluß angelegten Instruktionsdatensignalen, wobei der detektierte Substitutionsbefehl an den anderen Eingang des ODER-Gatters und das invertierte Signal des detektierten Substitutionsbefehls an den anderen Eingang des UND-Gatters angelegt wird.

3. Testmusterergenerator nach Anspruch 1 oder 2, wobei jeder der mehreren Registersätze in der Instruktionsverarbeitungsschaltung n Register in paralleler Form aufweist, wobei jeder Registersatz die Instruktionsdatensignale auf der Basis des Taktsignals an den nächsten Registersatz weiterschiebt.

Hierzu 11 Seite(n) Zeichnungen

- Leerseite -

This Page Blank (uspto)

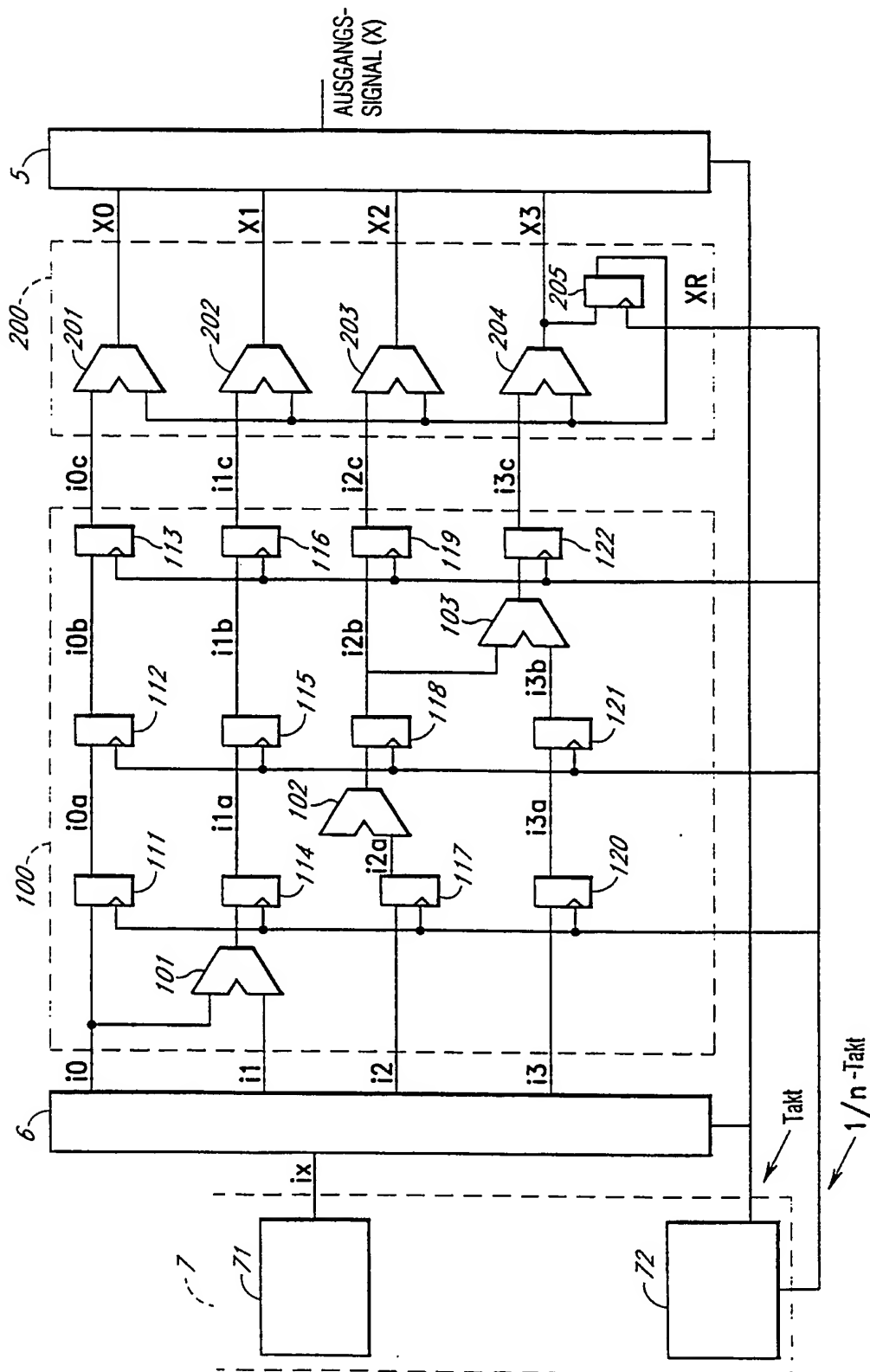


FIG. 1

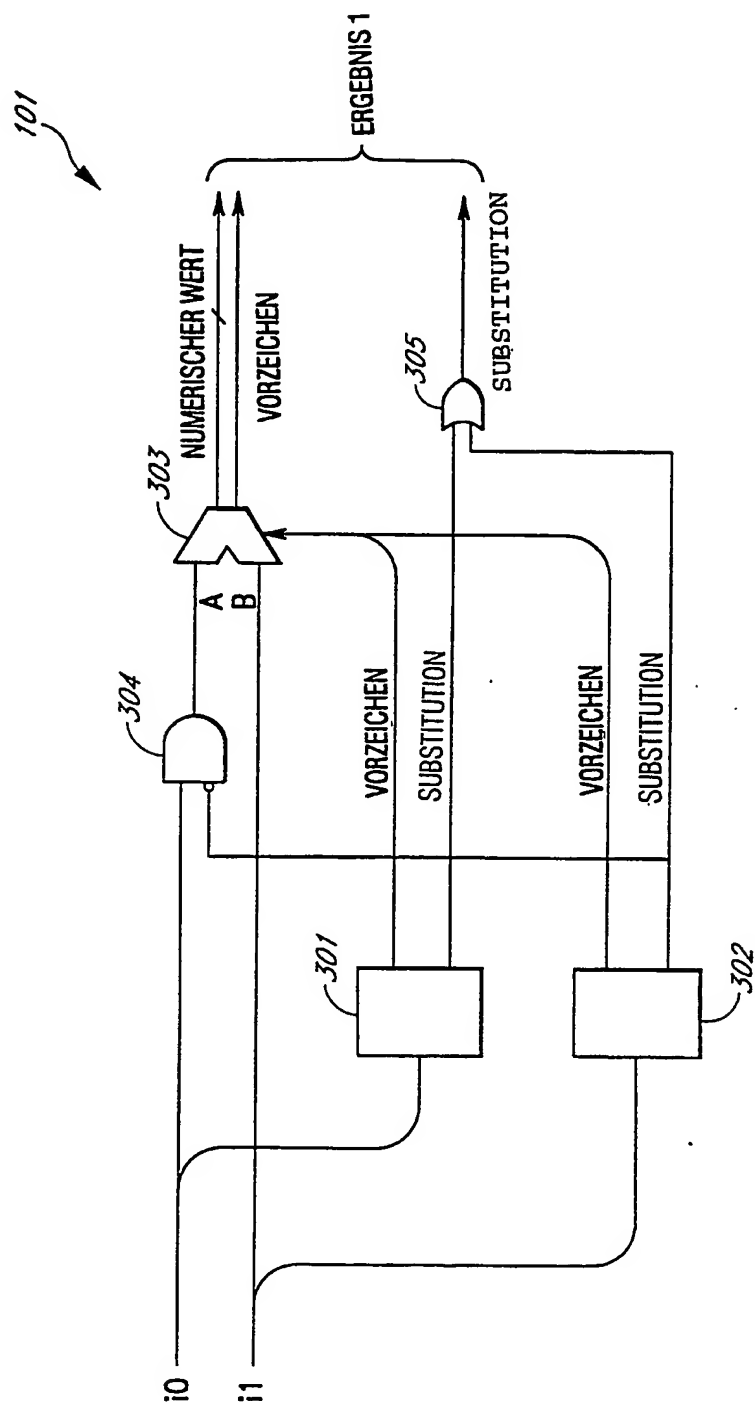
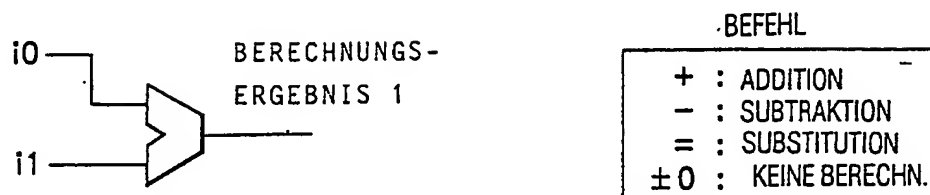


FIG. 2



i0	i1	ERGEBNIS 1
+a	+b	$+(a+b)$
+a	-b	$+(a-b) : a > b$ $-(b-a) : a < b$
+a	=b	$=(b)$
+a	± 0	$+(a)$
-a	-b	$-(a+b)$
-a	+b	$-(a-b) : a > b$ $+(b-a) : a < b$
-a	=b	$=(b)$
-a	± 0	$-(a)$
=a	+b	$=(a+b)$
=a	-b	$=(a-b)$
=a	=b	$=(b)$
=a	± 0	$=(a)$
± 0	+b	$+(b)$
± 0	-b	$-(b)$
± 0	=b	$=(b)$
± 0	± 0	$±(0)$

FIG. 3

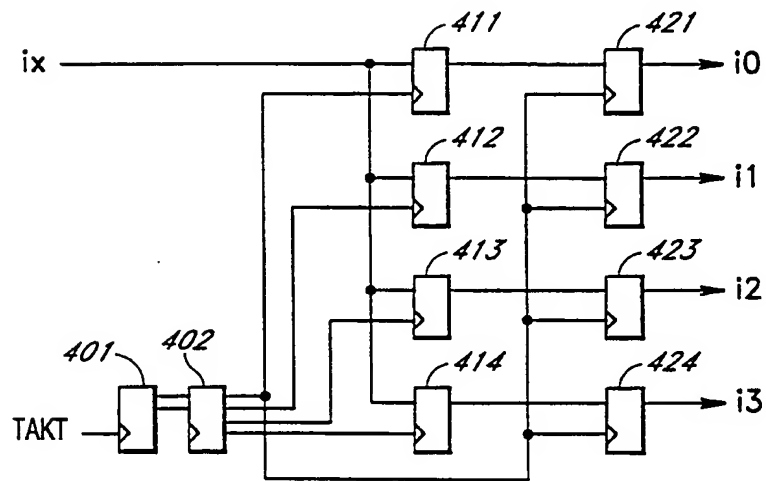


FIG. 4

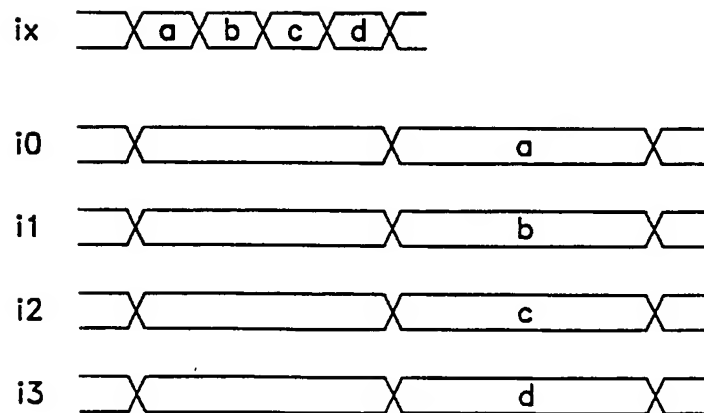


FIG. 5

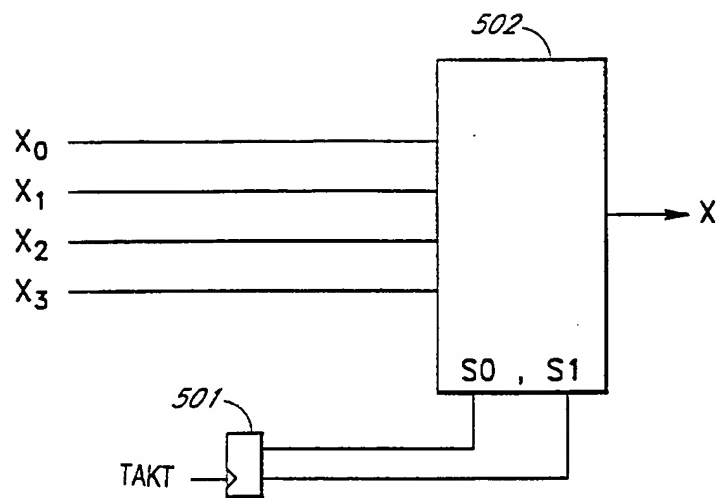


FIG. 6

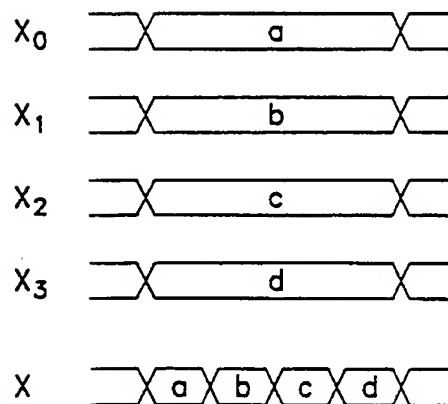


FIG. 7

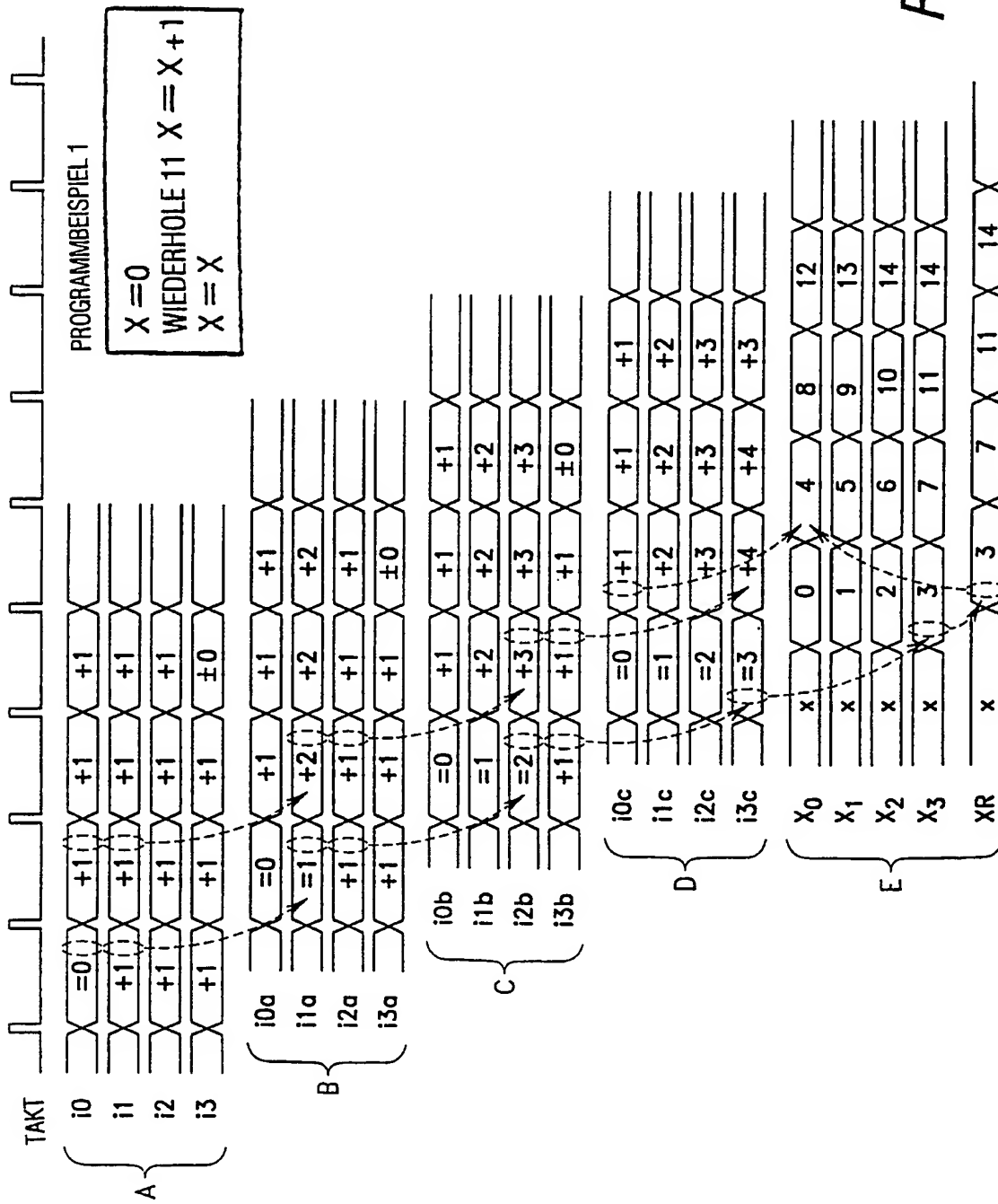
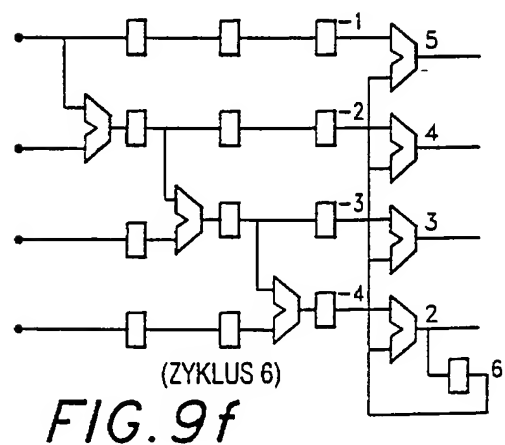
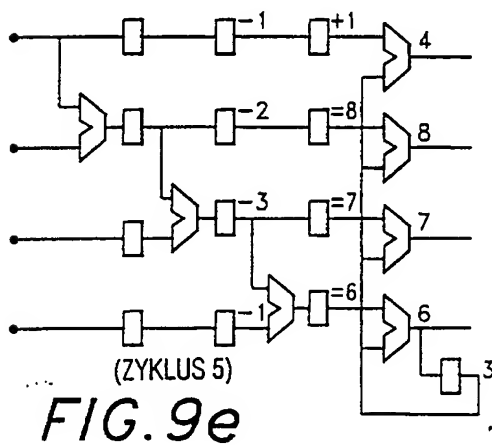
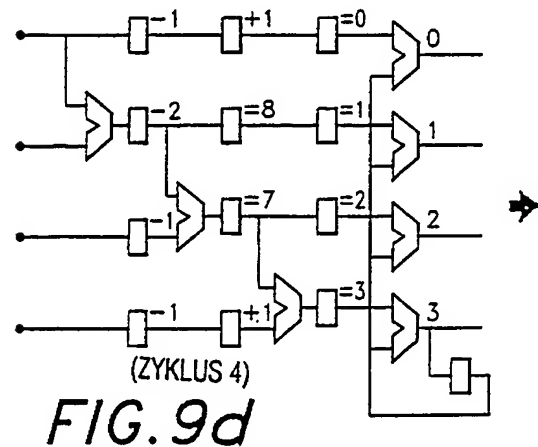
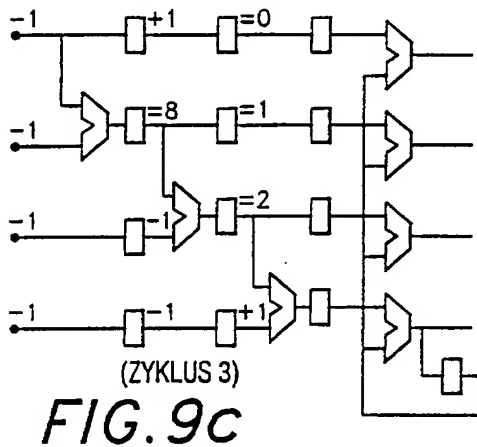
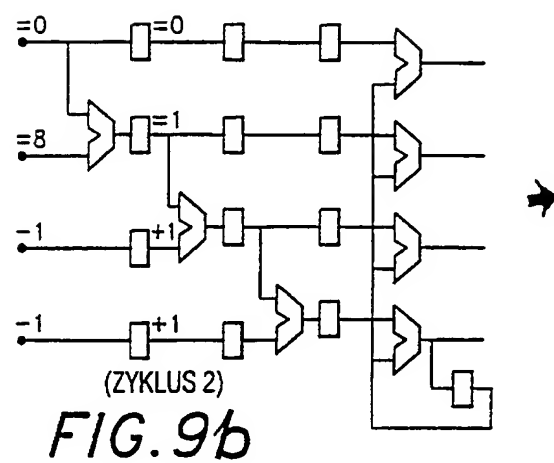
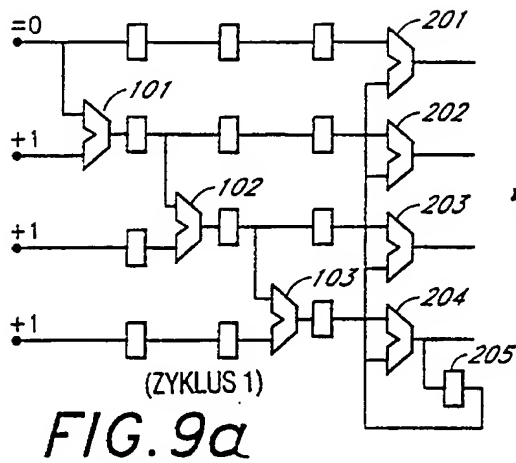


FIG. 8



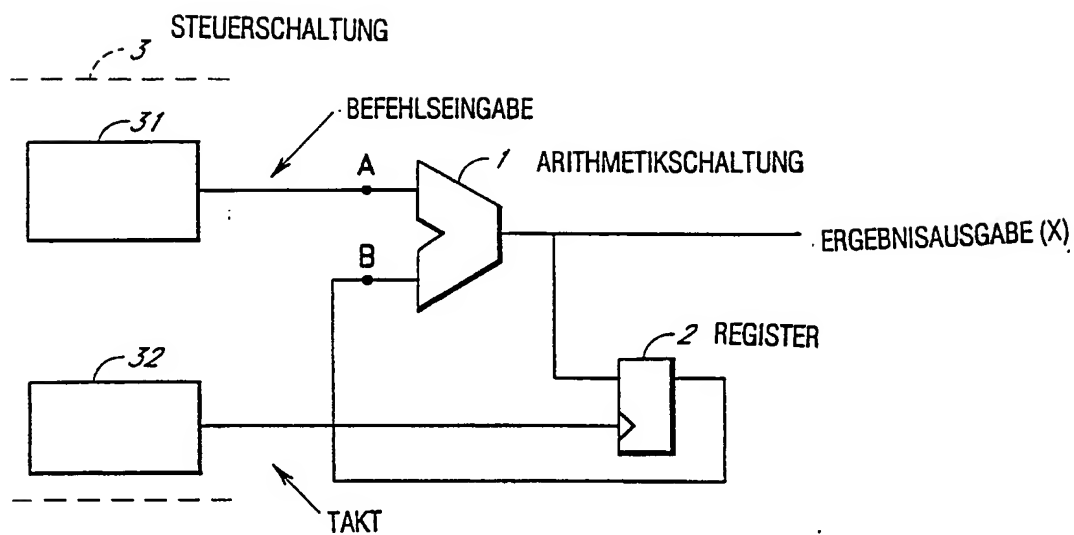


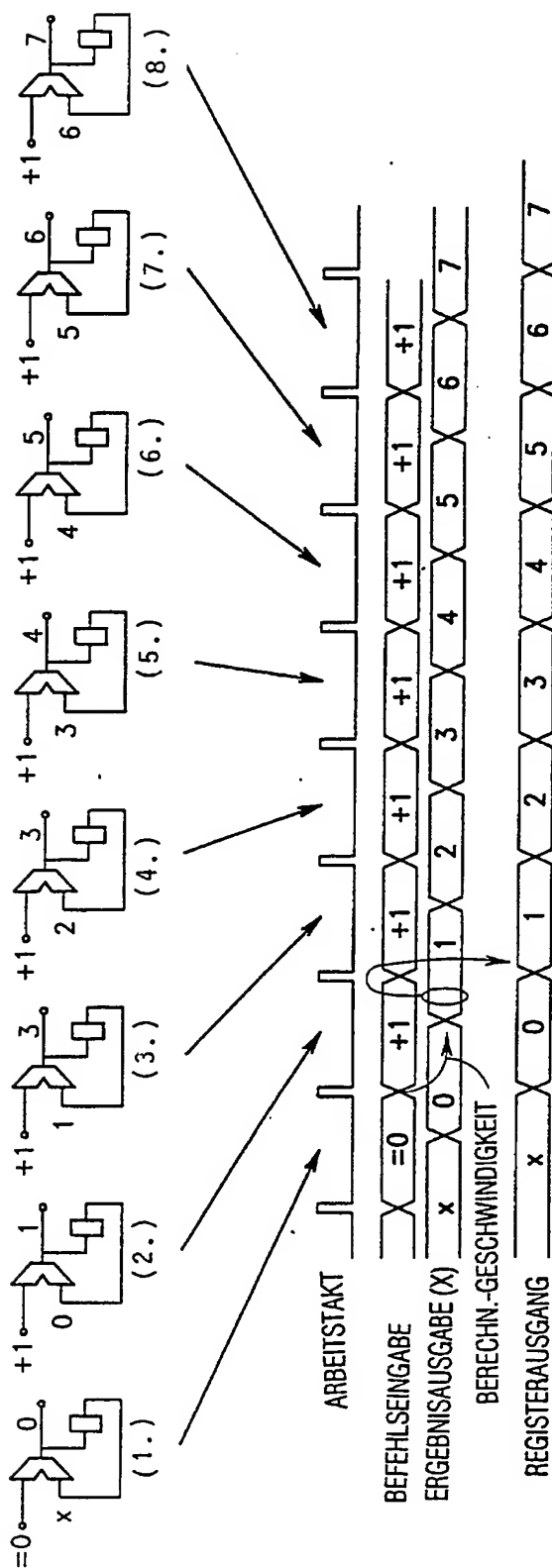
FIG. 10
STAND DER TECHNIK

FIG. 11

STAND DER TECHNIK

PROGRAMMBEISPIEL 1

X=0
WIEDERHOLE 11 X=X+1
X=X



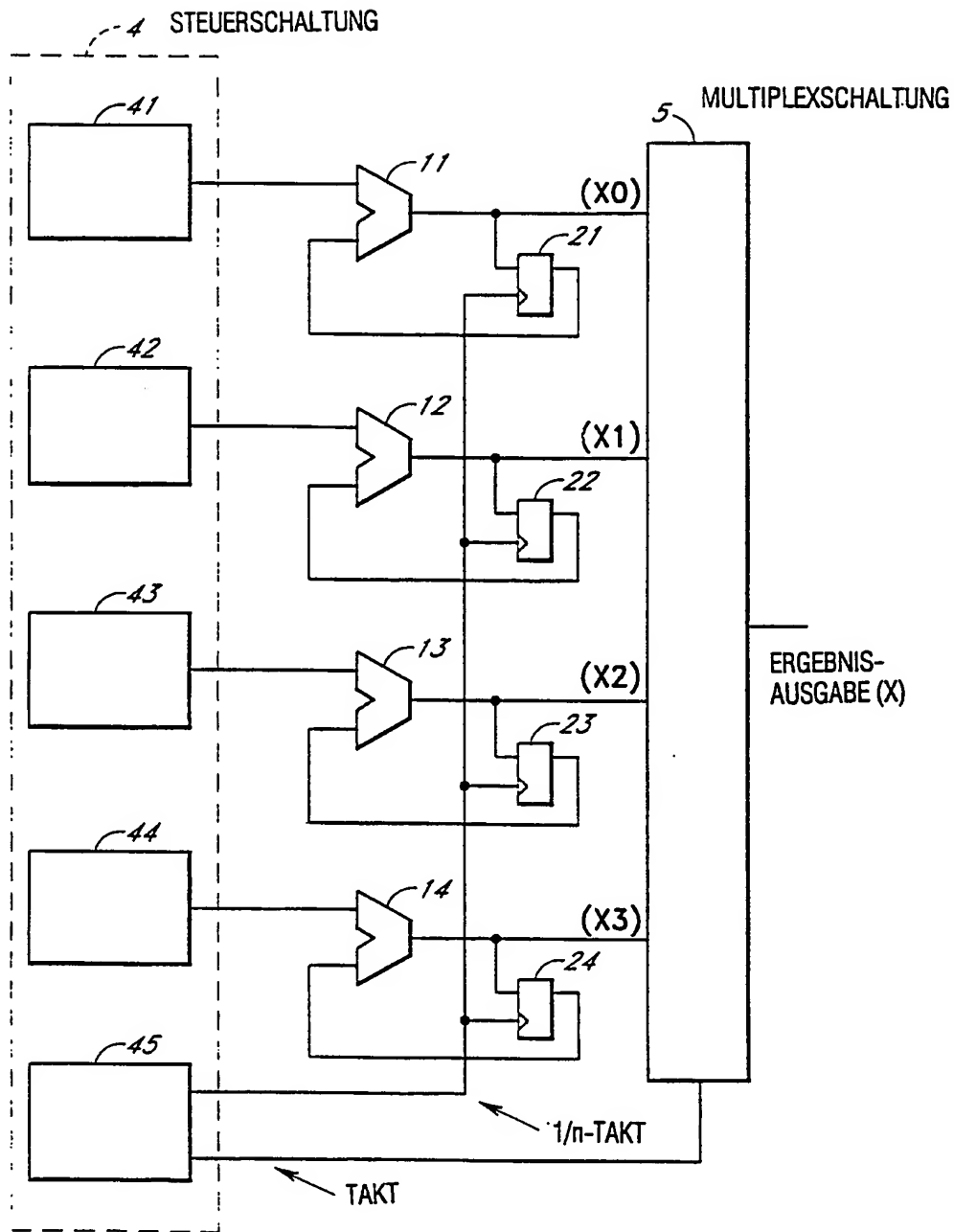


FIG. 12
STAND DER TECHNIK

PROGRAMMBEISPIEL 1

$X_0 = 0$ WIEDERHOLE 2 $X_0 = X_0 + 4$
$X_1 = 1$ WIEDERHOLE 2 $X_1 = X_1 + 4$
$X_2 = 2$ WIEDERHOLE 2 $X_2 = X_2 + 4$
$X_3 = 3$ $X_3 = X_3 + 4$ $X_3 = X_3 + 3$

FIG. 13A
STAND DER TECHNIK

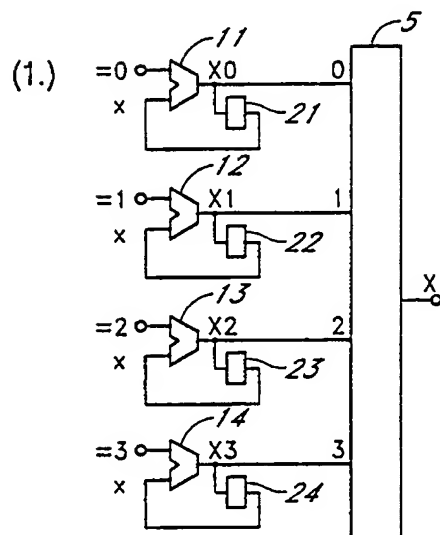


FIG. 13b

STAND DER TECHNIK

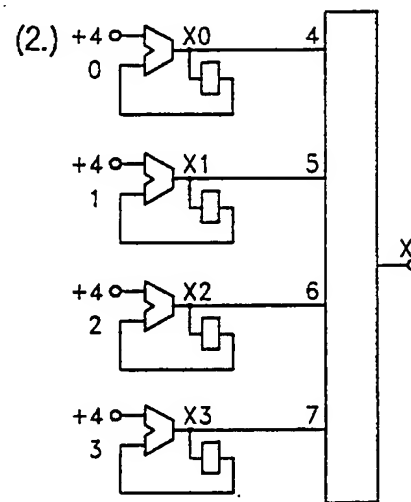


FIG. 13c

STAND DER TECHNIK

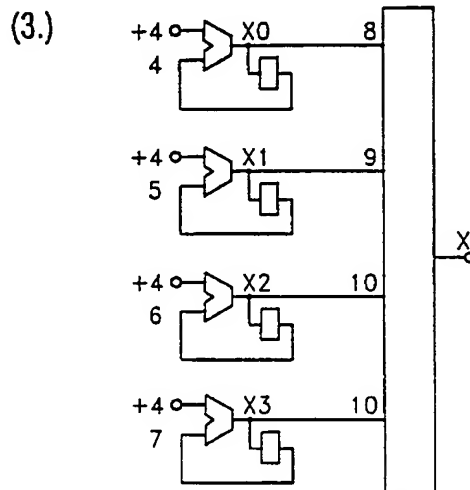


FIG. 13d

STAND DER TECHNIK